



Differential algebra and system modeling in cellular biology

François Boulrier, François Lemaire

► To cite this version:

François Boulrier, François Lemaire. Differential algebra and system modeling in cellular biology. AB 2008, 2008, Hagenberg, Austria. pp.22-39. hal-00274689

HAL Id: hal-00274689

<https://hal.science/hal-00274689>

Submitted on 21 Apr 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Differential algebra and system modeling in cellular biology

François Boulier and François Lemaire

University Lille I, LIFL, 59655 Villeneuve d'Ascq, France,
{Francois.Boulier,Francois.Lemaire}@lifl.fr,
<http://www.lifl.fr/~{boulier,lemaire}>

Abstract. Among all the modeling approaches dedicated to cellular biology, differential algebra is particularly related to the well-established one based on nonlinear differential equations. In this paper, it is shown that differential algebra makes both simple and algorithmic one of the model reduction methods, the quasi-steady state approximation theory, in the particular setting of generalized chemical reactions systems. This recent breakthrough may suggest some evolution of modeling techniques based on nonlinear differential equations, by incorporating the reduction hypotheses in the models. Potential improvements of parameters fitting methods are discussed too.

1 Introduction

Among all the modeling approaches dedicated to cellular biology, differential algebra is particularly related to the well-established one based on nonlinear differential equations [1–3].

Differential equations apply to rather small models. There are many subapproaches, restricted to (sometimes piecewise) linear differential equations, qualitative simulations ... [4–6]. The approach based on nonlinear differential equations was however successfully applied for the analysis of many genetic regulatory circuits, including those that work during complex animal embryogenesis [7]. One of the major successes of this approach is certainly the modeling of the segment polarity network of the drosophila [8, 9].

This paper is mostly concerned by the model reduction problem for chemical reactions systems based on the generalized mass action law [10]. Among all the approximation techniques available for such systems [11, lumping, sensitivity analysis, ...], this paper is exclusively concerned by the quasi-steady state approximation theory.

The model reduction problem for systems of differential equations broadly consists in simplifying the given system, by means of some simplification hypotheses. It aims to get a tractable reduced system, while preserving some properties of interest (e.g. presence of oscillations, number of equilibria).

Modeling approaches based on nonlinear differential equations do not all directly rely on the formalism of chemical reactions systems. Quite often, more

sophisticated functions are used such as the Henri-Michaelis-Menten factors and the Hill functions. It is worth noticing that these sophisticated functions can quite often be deduced from generalized chemical reactions systems through some model reduction. However, practitioners using these functions do not always explicitly formulate the simplifying assumptions which justify these approximations. Thereby, they do not formulate the domains of validity of their models as precisely as they could.

Differential algebra makes both simple and algorithmic the quasi-steady state approximation methods in the particular setting of systems of generalized chemical reactions systems [12]. This fact thus suggests to widen the use of these systems for systems modeling, and to incorporate in the models the hypotheses which lead to the more sophisticated formulas. One should thereby expect to obtain models with more ascertained domains of validity and to estimate the practical relevance of some of the simplification hypotheses.

In addition to the above issues, the differential algebra and the quasi-steady state approximation theory also suggest improvements of the parameters fitting problem, which is an important concern in systems modeling. It has been known for a few years that differential elimination somehow transforms nonlinear least squares into linear ones in the parameters estimation problem, by suggesting a starting point for the Newton-like estimation methods [13, 14]. By incorporating moreover, in these methods, the recent progress obtained in the quasi-steady state approximation theory, one may also expect to reduce the stiffness of the differential systems that they need to numerically integrate, thereby speeding up the overall optimisation processes.

2 Differential elimination

The content of this section owes a lot to [15]. We refer to that paper for more details.

The differential elimination theory is a subtheory of the differential algebra [16, 17]. See also [18]. The differential elimination processes that are presented in this paper take as input two parameters: a system of polynomial (thus nonlinear) differential equations, ordinary or with partial derivatives¹ and a *ranking*². They produce on the output an equivalent finite set of polynomial differential systems, which are simpler, in the sense that they involve some differential equations which are consequences of the input system but were somehow hidden. The output may consist of more than one differential system because the differential elimination process may need to split cases. The set of the differential equations which are consequences of the input system forms a so-called *differential ideal* of some *polynomial differential ring*. Since this ideal is an infinite set, a natural question arises: how does the process select the finitely many differential equations which appear in the output system ? This is indeed the role of the rankings.

¹ This paper is only concerned by the ordinary case.

² Emphasized words have a technical meaning, which will be defined in section 2.2.

2.1 An example

The following example, borrowed from [19, Chapter VII, page 454], is a so-called differential algebraic equations system, since it mixes differential and non differential equations. There are three unknown time varying functions (three dependent variables) x , y and z . The dot over a variable denotes its derivative w.r.t. the independent variable, which is assumed to be the time, throughout this paper :

$$\begin{aligned}\dot{x} &= 0.7 y + \sin(2.5 z), \\ \dot{y} &= 1.4 x + \cos(2.5 z), \\ 1 &= x^2 + y^2.\end{aligned}$$

Let us try to apply an explicit numerical integration scheme such as Euler's one over this system, for some initial conditions. For each dependent variable (say) x , the scheme transforms the explicit differential equation $\dot{x} = f(x, y, z)$ as a recurrence formula $x_{n+1} = x_n + h f(x_n, y_n, z_n)$, where h denotes some stepsize, and computes the terms x_n the ones after the others. However, the absence of a differential equation of the following form seems to forbid the application of Euler's scheme:

$$\dot{z} = f(x, y, z). \quad (1)$$

Indeed, this equation is not missing. It is a "hidden" consequence of the input system: it can be automatically extracted from it by differential elimination. In order to show how to proceed with the help of the *diffalg* package of MAPLE, one first converts the system as a *polynomial differential system*. For this, one denotes s the sine, c the cosine and one introduces a few more equations. The following differential polynomial system is equivalent to the above one.

$$\begin{aligned}\dot{x} &= 0.7 y + s, & \dot{s} &= 2.5 \dot{z} c, \\ \dot{y} &= 1.4 x + c, & \dot{c} &= -2.5 \dot{z} s, \\ 1 &= x^2 + y^2. & 1 &= s^2 + c^2.\end{aligned}$$

In order to compute equation (1) using *diffalg*, one stores the differential polynomial system in the variable *syst*, converting floating point numbers as rational numbers.

```
with (diffalg):
syst := [diff(x(t),t) - 7/10*y(t) - s(t),
         diff(y(t),t) - 14/10*x(t) - c(t),
         x(t)^2 + y(t)^2 - 1,
         diff(s(t),t) - 25/10*diff(z(t),t)*c(t),
         diff(c(t),t) + 25/10*diff(z(t),t)*s(t),
         s(t)^2 + c(t)^2 - 1]:
```

Then the variable R is assigned the context of the computation: the only derivation is taken with respect to the time, the notation is the standard *diff* notation of MAPLE and the *ranking* is provided. According to the ranking notation of the *diffalg* package, this is the *orderly* ranking such that $s > c > x > y > z$.

The properties of that ranking ensure that, if there exists a differential equation of the form (1) in the *radical* of the *differential ideal* generated by *syst* then the differential elimination process will find it.

```
R := differential_ring (derivations = [t], notation = diff,
                      ranking = [[s, c, x, y, z]]):
```

Next the *Rosenfeld-Gröbner* function [20] is applied to *syst* and *R*. It returns a list of MAPLE tables. Each table provides a *regular differential chain* defining some differential ideal. The list should be understood as an intersection. Over the example, the list only involves one entry so that the regular differential chain does represent the radical differential ideal generated by the input system. The desired equation stands on the second place of the chain (only the two first equations are displayed). Enlarging the input system with this equation, it is now easy to perform any numerical integration method and our problem is solved.

```
ideal := Rosenfeld_Groebner (syst, R):
rewrite_rules (ideal [1]):
```

$$\left[\begin{aligned} \frac{d}{dt}y(t) &= \frac{7}{5}x(t) + c(t), \\ \frac{d}{dt}z(t) &= \frac{1}{25} \frac{3500 - 12348(y(t))^6 + 13230c(t)x(t)(y(t))^4 + 25809(y(t))^4}{441(y(t))^6 - 882(y(t))^4 + 541(y(t))^2 - 100} \\ &\quad + \frac{1}{25} \frac{-14700x(t)(y(t))^2c(t) - 16961(y(t))^2 + 3940x(t)c(t)}{441(y(t))^6 - 882(y(t))^4 + 541(y(t))^2 - 100}, \quad \dots \end{aligned} \right]$$

2.2 Differential algebra

A *differential ring* (resp. field) is a ring (resp. field) R endowed with a derivation (this paper is restricted to the case of a single derivation but the theory is more general) i.e. a unitary mapping $R \rightarrow R$ such that (denoting \dot{a} the derivative of a):

$$\widehat{(a+b)} = \dot{a} + \dot{b}, \quad \widehat{(ab)} = \dot{a}b + a\dot{b}.$$

Observe that, theoretically, the derivation is an abstract operation. For legibility, one views it as the derivation w.r.t. the time t . Algorithmically, one is led to manipulate finite subsets of some *differential polynomial ring* $R = K\{U\}$ where K is the differential field of coefficients (in practice, $K = \mathbb{Q}, \mathbb{Q}(t)$ or $\mathbb{Q}(k_1, \dots, k_r)$ where the k_i denote parameters that would be assumed to be algebraically independent) and U is a finite set of *dependent variables*³. The elements of R , the

³ In the differential algebra theory, the terminology *differential indeterminates* is preferred to *dependent variables* for derivations are abstract and differential indeterminates are not even assumed to correspond to functions. In order not to mix different expressions in this paper, the second expression, which seems to be more widely known, was chosen.

differential polynomials are just polynomials in the usual sense, built over the infinite set, denoted ΘU , of all the derivatives of the dependent variables.

Definition 1. A differential ideal of a differential ring R is an ideal of R , stable under the action of the derivation.

Let F be a finite subset of a differential ring R . The set of all the finite linear combinations of various orders derivatives of elements of F , with elements of R for coefficients, is a differential ideal. It is called the differential ideal generated by F . An ideal \mathfrak{A} is said to be *radical* if $a \in \mathfrak{A}$ whenever there exists some nonnegative integer p such that $a^p \in \mathfrak{A}$. The radical of an ideal \mathfrak{A} is the set of all the ring elements a power of which belongs to \mathfrak{A} . The radical of a (differential) ideal is a radical (differential) ideal.

Theorem 1. Let R be a differential polynomial ring and F be a finite subset of R . A differential polynomial p of R lies in the radical of the differential ideal generated by F if and only if it vanishes over every analytic solution of F .

Proof. [16, chap. II, §7, 11] or [21].

The *Rosenfeld-Gröbner* algorithm [20] solves the membership problem to radical differential ideals. To present it, one needs to define the concept of *ranking*.

Definition 2. If U is a finite set of dependent variables, a ranking over U is a total ordering over the set ΘU of all the derivatives of the elements of U which satisfies: $a < \dot{a}$ and $a < b \Rightarrow \dot{a} < \dot{b}$ for all $a, b \in \Theta U$.

Let U be a finite set of dependent variables. A ranking such that, for every $u, v \in U$, the i th derivative of u is greater than the j th derivative of v whenever $i > j$ is said to be *orderly*. If U and V are two finite sets of dependent variables, one denotes $U \gg V$ every ranking such that any derivative of any element of U is greater than any derivative of any element of V . Such rankings are said to *eliminate* U w.r.t. V .

Assume that some ranking is fixed. Then one may associate with any differential polynomial $f \in K\{U\} \setminus K$ the greatest (w.r.t. the given ranking) derivative $v \in \Theta U$ such that $\deg(f, v) > 0$. This derivative is called the *leading derivative* or the *leader* of f .

Rankings permit to define leaders. Leaders permit to use differential polynomial as rewrite (substitution) rules. Assume that $f = a_d v^d + \dots + a_1 v + a_0$ is a differential polynomial with leader v (the coefficients a_i are themselves differential polynomials). Then the equation $f = 0$ can be written (as the *rewrite_rules* function of *diffalg* presented in section 2.1 does, though it uses an equal sign instead of an arrow):

$$v^d \longrightarrow -\frac{a_{d-1} v^{d-1} + \dots + a_1 v + a_0}{a_d}.$$

It can be used afterwards as a rule to simplify any differential polynomial g such that $\deg(g, v) \geq d$ or $\deg(g, v^{(k)}) > 0$ where $v^{(k)}$ denotes any proper derivative

of v . There are precise algorithms for performing these sorts of substitution by finite sets of rewrite rules: Ritt’s reduction algorithm or the *normal form* algorithm [21, algorithm NF].

The *Rosenfeld-Gröbner* algorithm gathers as input a finite system F of differential polynomials and a ranking. It returns a finite family (possibly empty) C_1, \dots, C_r of finite subsets of $K\{U\} \setminus K$, called *regular differential chains*. Each system C_i defines a differential ideal \mathfrak{C}_i (it is a *characteristic set* of \mathfrak{C}_i) in the sense that, for any $f \in K\{U\}$, we have

$$f \in \mathfrak{C}_i \quad \text{iff} \quad \text{NF}(f, C_i) = 0.$$

The relationship with the radical \mathfrak{A} of the differential ideal generated by F is the following:

$$\mathfrak{A} = \mathfrak{C}_1 \cap \dots \cap \mathfrak{C}_r.$$

When $r = 0$ we have $\mathfrak{A} = K\{U\}$. Combining both relations, one gets an algorithm to decide membership in \mathfrak{A} . Indeed, given any $f \in K\{U\}$ we have:

$$f \in \mathfrak{A} \quad \text{iff} \quad \text{NF}(f, C_i) = 0, \quad 1 \leq i \leq r.$$

The differential ideals \mathfrak{C}_i do not need to be prime. They are however necessarily radical. The $\text{NF}(\cdot, C_i)$ function permits to compute canonical representatives of the residue classes of the differential ring R/\mathfrak{C}_i .

3 Quasi-steady state approximation for generalized chemical reactions systems

Differential elimination makes the quasi-steady state approximation (QSSA) theory both simple and algorithmic in the special setting of generalized⁴ chemical reactions systems as shown by [12].

3.1 QSSA in general: fast and slow variables

In principle, QSSA applies to systems under the two time-scales standard form. Consider the following system in two dependent variables x and y (for legibility, but there may be more than two variables) and assume that ε is a small positive number.

$$\dot{x} = f(x, y), \quad \varepsilon \dot{y} = g(x, y). \quad (2)$$

On a random point $(x, y) \in \mathbb{R}^2$, the speed of y is high and thus, under some general conditions, rapidly approaches an area where $g(x, y) \simeq 0$. The variable x is said to be *slow* while the variable y is said to be *fast*. The QSSA amounts to approximate such a system by the following one:

$$\dot{x} = f(x, y), \quad 0 = g(x, y) \quad (3)$$

⁴ Chemical reactions systems are said to be *generalized* when their elementary reactions are not required to be balanced. See [10].

which mixes differential and algebraic equations. Note that this approximation is only valid under some conditions (e.g. stability of the differential system in the neighborhood of $g(x, y) = 0$) given in the Tikhonov theorem [22] and after the transient step is elapsed.

Performing a QSSA over a differential system presents two advantages: it reduces the number of ODE and it tends to transform stiff systems into nonstiff ones, much easier to solve numerically.

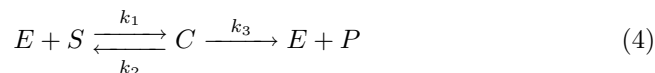
However, the QSSA is not proven algorithmic in general. The issue relies in the fact that the fast and slow variables, if they exist, may only be obtained through a change of coordinates and there does not seem to exist any algorithm which decides, given a differential system, if such a change of coordinates does exist.

3.2 QSSA for chemical reactions systems: fast and slow reactions

For differential systems arising from generalized chemical reactions systems, there exists a standard way to perform the QSSA, provided that the set of chemical reactions is divided in two parts: the fast ones and the slow ones.

As far as we know, the first clear relationship between this method and the Tikhonov theorem was established in [23]. Afterwards, close variants of the same method were rediscovered more or less independently [24, 25]. Though all these papers present methods, none of them is fully presented in an algorithmic manner. This may at least partly be due to the fact that some steps of the methods require the inversion of a matrix over a residue class ring, a non obvious task which may imply splitting cases.

Indeed, it turns out that the whole method is equivalent to a differential elimination process, as shown for the first time in [12]. For a general presentation of the method, one refers to [12]. In this paper, one only presents the method over a famous example: the Henri-Michaelis-Menten reduction of the basic enzymatic reaction system:



The initial system of ODE writes: $\dot{X} = N V$ i.e.

$$\begin{pmatrix} \dot{E} \\ \dot{C} \\ \dot{S} \\ \dot{P} \end{pmatrix} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} k_1 E S \\ k_2 C \\ k_3 C \end{pmatrix} \quad (5)$$

where X is the vector of the chemical species, N is the system stoichiometry matrix and V is the vector of the reaction rates. The stoichiometry matrix is built as follows: it involves one row per species and one column per reaction. The entry at row r , column c is the number of molecules of species r produced by the reaction c (i.e. the number of times species r occurs on the reaction right

handside minus the number of times it occurs on the reaction left handside). The rate of a reaction is the product of the left handside species (with multiplicities) times the reaction rate (the parameter over the arrow). Expanding the formula, one gets:

$$\begin{aligned}\dot{E} &= -k_1 E S + (k_2 + k_3) C, \\ \dot{S} &= -k_1 E S + k_2 C, \\ \dot{C} &= k_1 E S - (k_2 + k_3) C, \\ \dot{P} &= k_3 C.\end{aligned}\tag{6}$$

Among all the assumptions leading to the Henri-Michaelis-Menten formula:

$$\dot{S} = -\frac{V_{\max} S}{K + S},\tag{7}$$

the only one concerning the QSSA is that $k_1, k_2 \gg k_3$ i.e. that the two leftmost reactions of the chemical system (4) are fast while the rightmost one is slow. The other assumptions will be given later.

In order to perform the QSSA over the above system, one builds a DAE system from system (6) by replacing the contribution of the fast reactions by a new dependent variable, F_1 , and by inserting an algebraic equation, defining the “slow variety”, stating that the fast reactions are at quasi-equilibrium.

$$\begin{aligned}\dot{E} &= -F_1 + k_3 C \\ \dot{S} &= -F_1 \\ \dot{C} &= F_1 - k_3 C \\ \dot{P} &= k_3 C \\ 0 &= k_1 E S - k_2 C.\end{aligned}\tag{8}$$

In general there may be many different new dependent variables F_i and many different algebraic equations. Observe that, over the example, the two fast reactions are considered as one reversible reaction and associated to only one dependent variable: F_1 . The general process is described in section 3.3. Differential elimination is performed below using *diffalg*. The parameters are put in the base field \mathcal{F} of the equations to avoid discussions over their values (they are then considered as algebraically independent). The inequation $C(t) \neq 0$ is inserted to avoid considering this degenerate case. The output is pretty printed.

```
with(diffalg):
DAE_syst := [
    diff(E(t),t) - (-F1(t) + k3*C(t)),
    diff(S(t),t) + F1(t),
    diff(C(t),t) - (F1(t) - k3*C(t)),
    diff(P(t),t) - k3*C(t),
    k1*E(t)*S(t) - k2*C(t) ]:
F := field_extension (transcendental_elements=[k1,k2,k3]):
R := differential_ring (ranking=[[F1], [C,E,P,S]], notation=diff,
    derivations=[t], field_of_constants=F):
ideal := Rosenfeld_Groebner (DAE_syst, [C(t)], R):
    ideal := [characterizable]
rules := rewrite_rules(ideal [1]);
```

$$\left[\begin{aligned} F_1 &= \frac{k_3 k_1 E S (k_1 S + k_2)}{k_2 (k_1 S + k_1 E + k_2)}, & \dot{E} &= \frac{k_1^2 E^2 k_3 S}{k_2 (k_1 S + k_1 E + k_2)}, & \dot{P} &= \frac{k_3 k_1 E S}{k_2}, \\ \dot{S} &= -\frac{k_3 k_1 E S (k_1 S + k_2)}{k_2 (k_1 S + k_1 E + k_2)}, & C &= \frac{k_1 E S}{k_2} \end{aligned} \right]$$

Of course, one does not recognize formula (7) in this regular differential chain: the reduction is incomplete since the extra assumptions made in the Henri-Michaelis-Menten reduction have not yet been taken into account. These assumptions are: $S(0) \gg E(0)$, $P \simeq 0$ and $C(0) = 0$. One refers to [12] for a complete reduction.

3.3 Construction of the DAE to consider for the QSSA

First split the stoichiometry matrix N into two matrices N_f and N_s putting the columns which correspond to fast reactions in N_f and the ones which correspond to slow reactions in N_s . Split accordingly the rows of the vector V into two vectors V_f and V_s . One gets a formula $\dot{X} = N_s V_s + N_f V_f$. Over system (5), one gets:

$$\begin{pmatrix} \dot{E} \\ \dot{C} \\ \dot{S} \\ \dot{P} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 1 \end{pmatrix} \cdot (k_3 C) + \begin{pmatrix} -1 & 1 \\ 1 & -1 \\ -1 & 1 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} k_1 E S \\ k_2 C \end{pmatrix}. \quad (9)$$

Determine a maximal linearly independent set of columns of N_f (i.e. a basis of that matrix) and remove the other ones, giving a new matrix \bar{N}_f . Update the vector of reaction rates V_f , giving a new vector \bar{V}_f such that $N_f V_f = \bar{N}_f \bar{V}_f$. Over the example, removing the second column, one gets a new formula $\dot{X} = N_s V_s + \bar{N}_f \bar{V}_f$ giving system (6):

$$\begin{pmatrix} \dot{E} \\ \dot{C} \\ \dot{S} \\ \dot{P} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 1 \end{pmatrix} \cdot (k_3 C) + \begin{pmatrix} -1 \\ 1 \\ -1 \\ 0 \end{pmatrix} \cdot (k_1 E S - k_2 C). \quad (10)$$

Replace the vector \bar{V}_f by a vector F of new dependent variables F_i . The slow variety is defined by letting the entries of \bar{V}_f all equal to zero. The DAE to be considered for quasi-steady state approximation is

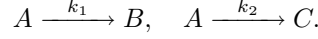
$$\dot{X} = N_s V_s + \bar{N}_f F, \quad \bar{V}_f = 0.$$

Over the example, one gets a formula giving system (8):

$$\begin{pmatrix} \dot{E} \\ \dot{C} \\ \dot{S} \\ \dot{P} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 1 \end{pmatrix} \cdot (k_3 C) + \begin{pmatrix} -1 \\ 1 \\ -1 \\ 0 \end{pmatrix} \cdot (F_1), \quad (k_1 E S - k_2 C) = 0. \quad (11)$$

3.4 Limits and generalizations of the method

In some cases, the F_i variables cannot be all eliminated. In that case, the method does not apply. In [12], this is checked by testing whether $\text{NF}(\dot{X}_i, C)$ only depends on the X_j (with order 0), for each dependent variable X_i and each regular differential chain C produced by differential elimination. A simple example is given by the generalized chemical reactions system, assuming k_1, k_2 are both fast:



If one applies the method sketched above, one gets the following differential-algebraic system:

$$\dot{A} = -F_1 - F_2, \quad \dot{B} = F_1, \quad \dot{C} = F_2, \quad 0 = k_1 A, \quad 0 = k_2 A$$

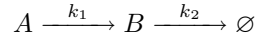
which simplifies to the regular differential chain:

$$F_1 = -\dot{C}, \quad F_2 = \dot{C}, \quad \dot{B} = -\dot{C}, \quad A = 0.$$

The normal form of \dot{B} is $-\dot{C}$ and \dot{C} is equal to its own normal form: the method failed.

According to the Tikhonov theorem, there are some extra conditions to check for the approximation to be valid [23, conditions C3 and C4]. In particular, the slow variety defined by the algebraic equations must be attractive.

In some cases, there exists a better slow variety than the one provided by the method. To compute a reduced system w.r.t. this different variety, one just needs to change the set of algebraic equations of the differential-algebraic system and to run the differential elimination process. Such a situation is illustrated by the next example borrowed from [26] where $k_2 \gg k_1$:



If one applies the method, one gets the following differential-algebraic system:

$$\dot{A} = -k_1 A, \quad \dot{B} = k_1 A - F_1, \quad 0 = k_2 B$$

which simplifies to the following regular differential chain:

$$F_1 = k_1 A, \quad \dot{A} = -k_1 A, \quad B = 0.$$

However, the slow variety $k_1 A - k_2 B = 0$ is better than $B = 0$. A better reduced system is thus obtained by performing differential elimination over the following differential-algebraic system:

$$\dot{A} = -k_1 A, \quad \dot{B} = k_1 A - F_1, \quad 0 = k_1 A - k_2 B$$

which simplifies to the following regular differential chain:

$$F_1 = (k_1 + k_2) B, \quad \dot{B} = -k_1 B, \quad A = \frac{k_2}{k_1} B.$$

4 Application to system modeling in cellular biology

This section aims at establishing some relationship between the quasi-steady state approximation method sketched in section 3 and system modeling in cellular biology. For this purpose, an application of this technique to the analysis of a genetic regulatory circuit made of a single gene, regulated by a polymer of its own protein [27, 28] is described.

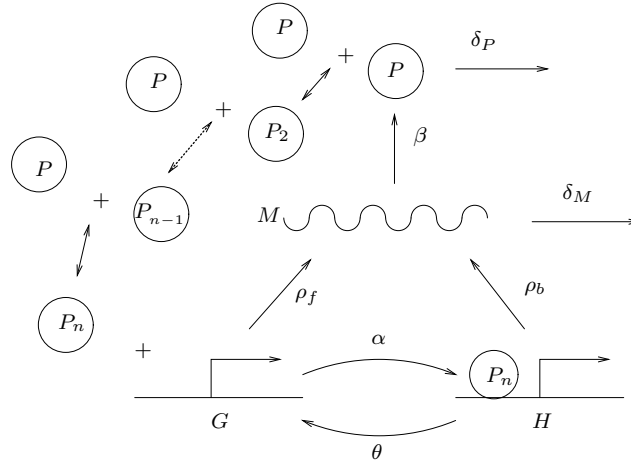
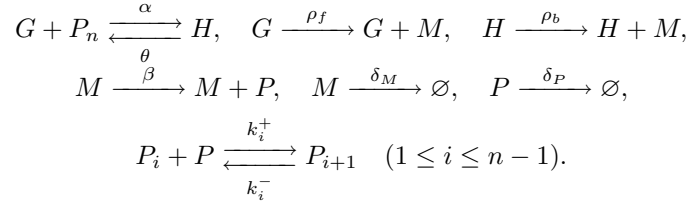


Fig. 1. A single gene regulated by a polymer of its own protein.

One considers the genetic circuit depicted in Figure 1. The single gene is regulated by an order n polymer of its own protein. The integer number n is a parameter of the system. This study was motivated by the activity of a working group aiming at modeling the circadian clock of the green alga *ostreococcus tauri*. The addressed question was: does there exist biologically meaningful parameters values which make this circuit oscillate ? More technically: does there exist biologically meaningful parameters values which make a Poincaré-Andronov-Hopf bifurcation occur ? One refers to [27] for a more detailed motivation of the addressed question and to [29] for a related work.

There are many different ways to derive a system of ordinary differential equations from the considered circuit but one of the simplest schemes consists in first translating it as a system of generalized chemical reactions (observe that transcription and translation are not balanced reactions). The variables G and H represent the state of the gene. The mRNA concentration and the concentration of the protein translated from the mRNA are represented by M and P . The n types of polymers of P are denoted by $P = P_1, P_2, \dots, P_n$. Greek letters and

k_i^-, k_i^+ ($1 \leq i \leq n-1$) represent parameters:



This generalized chemical reactions system can now be canonically translated as a system of parametric ordinary differential equations, denoting $A_i = (k_i^- P_{i+1} - k_i^+ P_i P)$. Variables $G, H, M, P = P_1, \dots, P_n$ are dependent variables. They all represent species concentrations except G and H , which should rather be viewed as “random variables”.

$$\begin{aligned}
\dot{G} &= \theta H - \alpha G P_n, \\
\dot{H} &= -\theta H + \alpha G P_n, \\
\dot{M} &= \rho_f G + \rho_b H - \delta_M M, \\
\dot{P} &= \beta M - \delta_P P + 2A_1 + A_2 + \dots + A_{n-1}, \\
\dot{P}_i &= -A_{i-1} + A_i & (2 \leq i \leq n-1), \\
\dot{P}_n &= -A_{n-1} + \theta H - \alpha G P_n
\end{aligned} \tag{12}$$

This system of $n+3$ differential equations depending on $2n+5$ parameters is actually much too large for any further symbolic analysis. It needs to be reduced.

In order to apply a quasi-steady state approximation, it is assumed that the $n-1$ chemical reactions describing the polymerization of the protein are fast compared to the other ones.

Then, according to the technique sketched in section 3, one gets an approximation of system (12) by replacing each expression A_i by a new dependent variable F_i ($1 \leq i \leq n-1$) and by augmenting this system by the $n-1$ following algebraic equations:

$$0 = k_i^+ P P_i - k_i^- P_{i+1}, \quad (1 \leq i \leq n-1)$$

It is now sufficient to eliminate the F_i from the so obtained differential-algebraic system. Unfortunately, this cannot be performed by a standard differential elimination algorithm since the number of equations depends on the parameter n . However, such an algorithm can be applied for many different values of n and the general formula can be inferred.

$$\begin{aligned}
\dot{G} &= \theta H - \alpha K_{n-1} P^n G, \\
\dot{H} &= -\theta H + \alpha K_{n-1} P^n G, \\
\dot{M} &= \rho_b H + \rho_f G - \delta_M M, \\
\dot{P} &= \frac{n\theta H - n\alpha K_{n-1} P^n G - \delta_P P + \beta M}{\sum_{i=0}^{n-1} (i+1)^2 K_i P^i}
\end{aligned} \tag{13}$$

where $K_i = \frac{k_1^+ \cdots k_i^+}{k_1^- \cdots k_i^-}$ with the convention $K_0 = 1$.

The redundant equation describing the evolution of H can be removed and H can be replaced by $\gamma_0 - G$ in the three remaining equations, where γ_0 is some new parameter denoting the total quantity of gene. Some further exact reduction of the parameters set can moreover be performed and one is led to the three parametric ODE model [28, system (3)].

This three parametric ODE reduced system is simple enough to be tackled by the Hopf criterion (a variant of the Routh-Hurwitz one) and it was proven in [27, 28] that a Poincaré-Andronov-Hopf bifurcation occurs for biologically meaningful values of the parameters if and only if $n > 8$. One refers to those papers for a biological interpretation of this result.

Let us conclude this section by a few comments on the involved quasi-steady state approximation methods. When [27] was written, its authors did not know the particular way to perform the quasi-steady state approximation over an ODE system derived from a generalized chemical reactions system. Thus they applied a “handmade” technique inspired from the singular perturbation theory yielding a three ODE model and “solved” it. Integral curves obtained by numerical integration showed differences between the initial and the reduced system but this was expected. Later, the same authors understood the particular technique for generalized chemical reactions system and showed how it is related to differential elimination [12]. This technique, applied over the very same system in [28], produced (after a suitable change of coordinates) a slightly different reduced system: one gets the old [27] reduced system by clearing the denominator of the last ODE of system (13). By some reduction argument, the proof of the old paper could be applied to the new model. This somehow confirmed that the handmade reduction technique was good. However, comparisons of integral curves obtained by numerical integration showed that the new model is more accurate, in the sense that it has a wider domain of validity, than the old one.

5 Parameters estimation

The problem which is addressed in this section can be stated as follows: given a parametric ordinary differential system and files of measures for some of the dependent variables, estimate the values of the unknown parameters. In this context, differential elimination somehow transforms a nonlinear least squares problem into a linear one by guessing a starting point for a Newton-like method.

The approach described in this section was published in [30, 13]. It was implemented in a software based on the BLAD libraries [31, 14]. It is strongly related to the study of the identifiability of parametric differential systems, for which a huge literature is available [32–36].

To be honest, the method is not very convenient for system modeling in cellular biology for it requires pretty precise model equations as well as pretty accurate measures. In cellular biology, one usually does not know the model equations though the situation might be better for in vitro systems or synthetic

genes. But accurate measures are definitely not available in 2008. However, we chose to present the method anyway in this paper for two reasons: first, the use of differential elimination always improves the classical numerical one ; second, the quasi-steady state approximation technique presented in section 3 suggests another, new, improvement.

The content of this section owes a lot to [15] and one refers to this paper for more detailed explanations.

The problem is stated over an academic example. Consider the following system of parametric ordinary differential equations. There are two dependent variables x_1, x_2 and four parameters k_{12}, k_{21}, k_e, V_e . It could be derived from a chemical reactions system since it involves two linear exchanges and a Henri-Michaelis-Menten degradation term.

$$\begin{aligned}\dot{x}_1 &= -k_{12} x_1 + k_{21} x_2 - \frac{V_e x_1}{k_e + x_1}, \\ \dot{x}_2 &= k_{12} x_1 - k_{21} x_2.\end{aligned}\tag{14}$$

Assume that x_1 is observed i.e. that a file of measures is available for this dependent variable but that x_2 is not. For the sake of simplicity, assume moreover that $x_2(0) = 0$ and $k_e = 7$ are known. Issue: estimate the values of the three unknown parameters k_{12}, k_{21}, V_e .

There exists a purely numerical method to solve this problem. It is based on a nonlinear least squares method i.e. a Newton method. The idea is simple: pick random values for the three unknown parameters. Integrate numerically the differential system w.r.t. these values and compare the curve obtained by simulation with the file of measures. The error is defined as the sum, for all abscissas, of the squares of the ordinates differences between the two curves. The Newton method (a Levenberg-Marquardt scheme was applied in [14]) updates the values of the three unknown parameters if the error is considered as too large. It stops either if the error is small enough or if a stationary point is reached. However, as every Newton method, nonlinear least squares are very sensitive to the choice of the starting point (the initial random values) and are very likely to end up in a local minimum, with wrong parameters values.

By means of differential elimination, it is sometimes possible to compute a first estimate of the unknown parameters. This estimate is usually not very precise but can be used as a starting point for the Newton method.

The idea consists in eliminating the non observed variables in order to get a relation which only involves the observed variable x_1 , its derivatives up to any order and the parameters. Let us proceed with the help of *diffalg*. The *Rosenfeld-Gröbner* algorithm⁵ is applied over the model equations. The ranking $x_2 \gg x_1$ eliminates x_2 w.r.t. x_1 . The right-hand side of the first model equation is a rational fraction. It is decomposed as a numerator and a denominator. The

⁵ There are more efficient algorithms than *Rosenfeld-Gröbner* for performing this elimination since the input system already is a characteristic set w.r.t. some orderly ranking: one could apply a change of ranking algorithm [33, 37] which would avoid splitting cases.

numerator is stored in the list of the equations (first parameter to *Rosenfeld-Gröbner*). The denominator is stored in the list of the *inequations*. To avoid splitting cases on parameters values, one views them as algebraically independent elements of the base field of the differential polynomials.

```
eq1 := diff (x1(t),t) + k12*x1(t) - k21*x2(t) + Ve*x1(t)/(ke + x1(t));
eq2 := diff (x2(t),t) - k12*x1(t) + k21*x2(t);
K := field_extension (transcendental_elements = [k21, k12, ke, Ve]):
R := differential_ring
      (derivations = [t], notation = diff,
       field_of_constants = K, ranking = [x2, x1]):
ideal := Rosenfeld_Groebner ([numer (eq1), eq2], [denom (eq1)], R);
```

$ideal := [characterizable]$

The characteristic set *ideal* involves two polynomials. The one which does not involve x_2 is the second one, which is displayed below, slightly pretty printed. The expressions enclosed between square brackets are called “parameters blocks”.

$$\ddot{x}_1 (x_1 + k_e)^2 + [k_{12} + k_{21}] \dot{x}_1 (x_1 + k_e)^2 + [V_e] \dot{x}_1 k_e + [k_{21} V_e] x_1 (x_1 + k_e) = 0.$$

This equation tells us that the systems parameters are in principle uniquely defined. Indeed, assume that the function x_1 is known. Then so are its derivatives \dot{x}_1 and \ddot{x}_1 . These three functions can therefore be evaluated for three different values of the time t . The known parameter k_e can be replaced by its value. One thereby gets an exactly determined system of three linear equations whose unknowns are the parameters blocks. This system admits a unique solution. The values of the parameters blocks being fixed, it is obvious (over this example !) that the values of k_{12} , k_{21} and V_e also are uniquely defined. QED.

In practice, the function x_1 is known from a file of measures and one can try to numerically estimate the values of its first and its second derivative. If the measures are free of noise, the first derivative can be quite accurately estimated but this is usually not the case for the second derivative. To overcome these difficulties due to numerical approximations, one builds an overdetermined linear system that one solves by means of linear least squares.

The values of the blocks of parameters being known, one still has to recover the values of the parameters by solving the above algebraic system. Over this example, it is very easy. The obtained values can then be used as a starting point for the Newton method. Observe that one cannot guarantee that this starting point does actually lead to the global minimum.

There are however two more important difficulties to overcome.

There may exist algebraic relations between the parameters blocks. There is no such relation over the example. But assume, for the sake of the explanation, that the computed differential polynomial involves the three following blocks of parameters so that the third block is the product of the two first ones:

$$[V_e], \quad [k_{21}], \quad [V_e k_{21}].$$

There is no doubt that the numerical values produced during the resolution of the linear overdetermined system would not satisfy this relation. This would imply that the final algebraic system to solve in order to get the values of the parameters would be inconsistent.

The differential systems that the Newton method needs to numerically integrate at runtime may become stiff, forcing the numerical integrators to choose very small step sizes, thus slowing down the whole optimisation process. Indeed, in these nonlinear fitting methods, the parameters are the variables. The sets of large and small parameters have to change during the process and stiffness is often caused by the presence of different time scales in the differential systems.

A promising idea would consist, in the context of systems evolving from generalized chemical reactions systems, in applying the quasi-steady state approximation technique described in section 3. The differential system to be integrated could be replaced by a reduced one. The error computation of the Newton method could then be performed over the reduced system, speeding up the overall process.

6 Conclusion

As a conclusion, let us give some elements of answer to a quotation of Eric Ponder (a director of Long Island Biology Association), borrowed from [38, Postscript]: “work on the mathematics [for biology] seems to me to have developed along two equally unprofitable lines”.

The use of computer algebra methods for performing differential elimination seems very promising for enhancing (at least) software for system modeling in cellular biology. The recent progresses in the quasi-steady state approximation theory suggest to widen the use of generalized chemical reactions systems and make more explicit the simplification hypotheses which would otherwise justify the use of Henri-Michaelis-Menten or Hill terms. Already available standalone C libraries for differential elimination [31] should still simplify this evolution.

In turn, the search of applications in the field of system modeling in cellular biology also led to improvements in differential algebra. It is this work which pushed the authors to study of the quasi-steady state approximation theory in the context of generalized chemical reactions systems and which has also suggested a new improvement to the nonlinear least squares methods.

Last, these works are far from over. For instance, an important theoretical challenge consists in designing an algorithmic method for determining good slow varieties of generalized chemical reactions systems. The authors are involved in another, more concrete challenge: to develop a modeling software based on these ideas and to use it to design and analyze synthetic gene networks.

References

1. Leloup, J.C., Goldbeter, A.: Modeling the molecular regulatory mechanism of circadian rhythms in *Drosophila*. *Bioessays* **22** (2000) 84–93

2. Fall, C.P., Marland, E.S., Wagner, J.M., Tyson, J.J.: Computational Cell Biology. Volume 20 of Interdisciplinary Applied Mathematics. Springer Verlag (2002)
3. Conrad, E.D., Tyson, J.J.: Modeling Molecular Interaction Networks with Non-linear Differential Equations. In Szallasi, Z., Stelling, J., Periwé, V., eds.: System Modeling in Cellular Biology: From Concepts to Nuts and Bolts, Cambridge, Massachusetts: The MIT Press (2006) 97–124
4. de Jong, H.: Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. *Journal of Computational Biology* **9**(1) (2002) 67–103
5. de Jong, H., Geiselmann, J., Hernandez, C., Page, M.: Genetic Network Analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics* **19**(3) (2003) 336–344
6. de Jong, H., Ropers, D.: Qualitative Approaches to the Analysis of Genetic Regulatory Networks. In Szallasi, Z., Stelling, J., Periwé, V., eds.: System Modeling in Cellular Biology: From Concepts to Nuts and Bolts, Cambridge, Massachusetts: The MIT Press (2006) 125–147
7. Saka, Y., Smith, J.C.: A mechanism for the sharp transition of morphogen gradient interpretation in *Xenopus*. *BMC Dev. Biol.* **7**(47) (2007)
8. von Dassow, G., Meir, E., Munro, E.M., Odell, G.M.: The segment polarity network is a robust developmental module. *Nature* **406** (2000) 188–192
9. von Dassow, G., Meir, E.: Exploring modularity with dynamical models of gene networks. In Schlosser, G., Wagner, G.P., eds.: Modularity in Development and Evolution, University of Chicago Press (2003) 245–287
10. Horn, F., Jackson, R.: General mass action kinetics. *Archive for Rational Mechanics and Analysis* **47** (1972) 81–116
11. Okino, M.S., Mavrovouniotis, M.L.: Simplification of Mathematical Models of Chemical Reaction Systems. *Chemical Reviews* **98**(2) (1998) 391–408
12. Boulier, F., Lefranc, M., Lemaire, F., Morant, P.E.: Model Reduction of Chemical Reaction Systems using Elimination (2007) Presented at the international conference MACIS 2007, <http://hal.archives-ouvertes.fr/hal-00184558>.
13. Denis-Vidal, L., Joly-Blanchard, G., Noiret, C.: System identifiability (symbolic computation) and parameter estimation (numerical computation). In: Numerical Algorithms. Volume 34. (2003) 282–292
14. Boulier, F., Denis-Vidal, L., Henin, T., Lemaire, F.: LÉPISME. In: proceedings of the ICPSS conference. (2004) Submitted to the Journal of Symbolic Computation, <http://hal.archives-ouvertes.fr/hal-00140368>.
15. Boulier, F.: Differential Elimination and Biological Modelling. Radon Series on Computational and Applied Mathematics (Gröbner Bases in Symbolic Analysis) **2** (October 2007) 111–139 <http://hal.archives-ouvertes.fr/hal-00139364>.
16. Ritt, J.F.: Differential Algebra. Dover Publications Inc., New York (1950) http://www.ams.org/online_bks/coll133.
17. Kolchin, E.R.: Differential Algebra and Algebraic Groups. Academic Press, New York (1973)
18. Wang, D.: Elimination Practice: Software Tools and Applications. Imperial College Press, London (2003)
19. Hairer, E., Wanner, G.: Solving ordinary differential equations II. Stiff and Differential–Algebraic Problems. 2 edn. Volume 14 of Springer Series in Computational Mathematics. Springer–Verlag, New York (1996)
20. Boulier, F., Lazard, D., Ollivier, F., Petitot, M.: Representation for the radical of a finitely generated differential ideal. In: ISSAC’95: Proceedings of the 1995 international symposium on Symbolic and algebraic computation, New York, NY, USA, ACM Press (1995) 158–166 <http://hal.archives-ouvertes.fr/hal-00138020>.

21. Boulier, F., Lemaire, F.: A computer scientist point of view on Hilbert's differential theorem of zeros. Submitted to *Applicable Algebra in Engineering, Communication and Computing* (2007) <http://hal.archives-ouvertes.fr/hal-00170091>.
22. Kokotovic, P., Khalil, H.K., O'Reilly, J.: *Singular Perturbation Methods in Control: Analysis and Design*. Classics in Applied Mathematics 25. SIAM (1999)
23. Van Breusegem, V., Bastin, G.: Reduced order dynamical modelling of reaction systems: a singular perturbation approach. In: *Proceedings of the 30th IEEE Conference on Decision and Control*, Brighton, England (December 1991) 1049–1054
24. Vora, N., Daoutidis, P.: Nonlinear model reduction of chemical reaction systems. *AIChE Journal* **47**(10) (2001) 2320–2332
25. Bennet, M.R., Volfson, D., Tsimring, L., Hasty, J.: Transient Dynamics of Genetic Regulatory Networks. *Biophysical Journal* **92** (May 2007) 3501–3512
26. Maquet, F.: Master Research training report. University Lille I (2008) to appear.
27. Boulier, F., Lefranc, M., Lemaire, F., Morant, P.E., Ürgüplü, A.: On proving the absence of oscillations in models of genetic circuits. In H. Anai, K.H., Kutsia, T., eds.: *Proceedings of Algebraic Biology 2007*. Volume 4545 of LNCS., Springer Verlag Berlin Heidelberg (2007) 66–80 <http://hal.archives-ouvertes.fr/hal-00139667>.
28. Boulier, F., Lefranc, M., Lemaire, F., Morant, P.E.: Applying a rigorous quasi-steady state approximation method for proving the absence of oscillations in models of genetic circuits. In: *Submitted to Algebraic Biology 2008*. (2008) <http://hal.archives-ouvertes.fr/hal-00213327>.
29. Niu, W., Wang, D.: Algebraic Approaches to Stability Analysis of Biological Systems. *Mathematics in Computer Science* **1** (2008) 507–539
30. Noiret, C.: Utilisation du calcul formel pour l'identifiabilité de modèles paramétriques et nouveaux algorithmes en estimation de paramètres. PhD thesis, Université de Technologie de Compiègne (2000)
31. Boulier, F.: The BLAD libraries. <http://www.lifl.fr/~boulier/BLAD> (2004)
32. Walter, É.: *Identifiability of State Space Models*. Volume 46 of *Lecture Notes in Biomathematics*. Springer Verlag (1982)
33. Ollivier, F.: Le problème de l'identifiabilité structurelle globale : approche théorique, méthodes effectives et bornes de complexité. PhD thesis, École Polytechnique, Palaiseau, France (1990)
34. Diop, S., Fliess, M.: Nonlinear observability, identifiability, and persistent trajectories. In: *Proc. 30th CDC*, Brighton (1991) 714–719
35. Ljung, L., Glad, S.T.: On global identifiability for arbitrary model parametrizations. *Automatica* **30** (1994) 265–276
36. Sedoglavic, A.: A Probabilistic Algorithm to Test Local Algebraic Observability in Polynomial Time. *Journal of Symb. Comp.* **33**(5) (2002) 735–755
37. Boulier, F., Lemaire, F., Moreno Maza, M.: Computing differential characteristic sets by change of ordering. Technical report, Université Lille I (2007) Submitted to the *Journal of Symbolic Computation*. <http://hal.archives-ouvertes.fr/hal-00141095>.
38. Mishra, B.: *Algebra, Automata, Algorithms & Beyond*. *Le Matematiche* **LXIII**(1) (2008) 21–23